

## EUROPA Quick Start

1. [Create a NDDL model for the domain](#)
2. [Create a problem instance](#)
3. [Instantiate and run a Solver](#)
4. [Examine Results](#)
5. [Embed your model into an application](#)
6. [Examples](#)
7. [Next Steps](#)

This tutorial provides just enough information for you to create and run simple planning problems using EUROPA. Once you're familiar with the basics, see the [Documentation](#) page for more details.

EUROPA can be used to solve Constraint Programming, Scheduling and Planning problem. For the purposes of this tutorial we will use a simple planning problem.

These are the main steps that a EUROPA user will normally follow to solve a problem :

- Create a model of the problem using NDDL, EUROPA's modeling language
- Create a particular instance of the problem using NDDL
- Instantiate a solver and ask it to look for a solution
- Inspect the plan, possibly modify the problem and re-plan
- When the model is working as expected, probably embed it into a larger application

Now, for our example, let's assume that we will create a plan to turn a light bulb on and off every 10 minutes. For this this simple domain here is what we would do.

TODO: Have PSUI run LightMgr by default, indicate at each step how to see the results. TODO: User must have build EUROPA already and have PLASMA and PlanWorks checked out

## Create a NDDL model for the domain

The New Domain Description Language (NDDL) is a simple but powerful language used to describe both problem domains and problem instances in EUROPA

```
// Light.nddl

// TODO!!: rewrite this example in terms of states and actions

class LightMgr {
  predicate turnOn  {}
  predicate turnOff {}
}

/**
 * @brief Simple rules to force a repeated cycle
 */

LightMgr::turnOn
{
  lt(10,duration);
  meets (object.turnOff);
}
```

```

}

LightMgr::turnOff
{
    eq(duration,10);
    meets (object.turnOn);
}

```

TODO: describe this model line by line

## Create a problem instance

```

// Light-initial-state.nddl

LightMgr mgr = new LightMgr();
fact(mgr.turnOn initialCondition);
eq(initialCondition.start,0);
goal(mgr.turnOff g);
lt(20,g.start);

```

## Instantiate and run a Solver

TODO: describe how to do this in PSUI

## Examine Results

TODO: describe how to do this in PSUI

## Embed your model into an application

Sometimes this is all you may want to do, probably dump the results out to a file.

TODO: describe briefly how to embed into a Java application, provide link for detail of how to embed in Java and C++

## Examples

Now that we have cover the entire cycle with a simple example, see the [Examples](#) page to see more advanced examples that can serve as starting points for your own model, or just to learn more about what EUROPA can do.

## Next Steps

TODO: provide links

- How to create your own project in PSUI
- How to embed your model into a C++ or Java application

- All the details about modeling in NDDL, configuring the built-in solver and using EUROPA's debugging tools.
- Extending EUROPA adding your own constraints, solvers, etc